

SQL FOR DATA ANALYSIS FULL PORTFOLIO PROJECT

REPORT ON PIZZA SALES

Introduction:

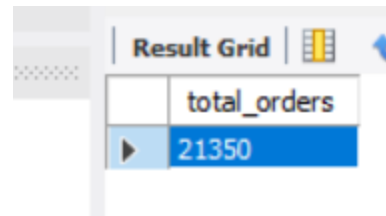
Hi, I'm Saranya Gelli. In this project I have utilized SQL queries to solve questions that were related to pizza sales

Questions solved:

- ▶ Retrieve the total number of orders placed.
- ▶ Calculate the total revenue generated from pizza sales.
- ▶ Identify the highest-priced pizza.
- ▶ Identify the most common pizza size ordered.
- ▶ List the top 5 most ordered pizza types along with their quantities.
- ▶ Join the necessary tables to find the total quantity of each pizza category ordered.
- ▶ Determine the distribution of orders by hour of the day.
- ▶ Join relevant tables to find the category-wise distribution of pizzas.
- ▶ Group the orders by date and calculate the average number of pizzas ordered per day.
- ▶ Determine the top 3 most ordered pizza types based on revenue.

Q1. Retrieve the total number of orders placed.

```
select * from orders;  
select count(order_id) as total_orders from orders;
```



A screenshot of a database query result grid. The grid has a header row with the column name 'total_orders' and a data row with the value '21350'. The grid is titled 'Result Grid' and has a small icon of a grid and a refresh button.

| total_orders |
|--------------|
| 21350 |

Q2. Calculate the total revenue generated from pizza sales.

```
select  
round(sum(order_details.quantity * pizzas.price),2) as total_sales  
from order_details join pizzas  
on pizzas.pizza_id = order_details.pizza_id;
```

| Result Grid | |
|-------------|-------------|
| | total_sales |
| ▶ | 817860.05 |

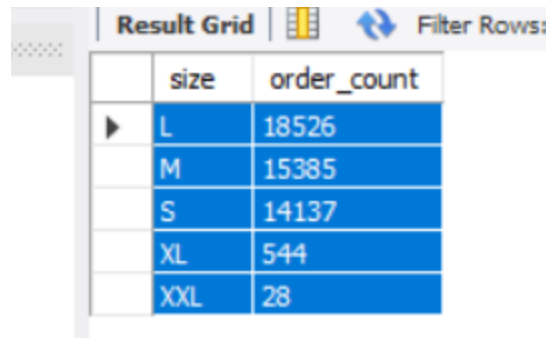
Q3. Identify the highest-priced pizza.

```
select pizza_types.name, pizzas.price
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
order by pizzas.price desc limit 1;
```

| Result Grid | | | Filter Rows: |
|-------------|-----------------|-------|--------------|
| | name | price | |
| ▶ | The Greek Pizza | 35.95 | |

Q4. Identify the most common pizza size ordered.

```
select pizzas.size, count(order_details.order_details_id) as order_count
from pizzas join order_details
on pizzas.pizza_id = order_details.pizza_id
group by pizzas.size order by order_count desc;
```

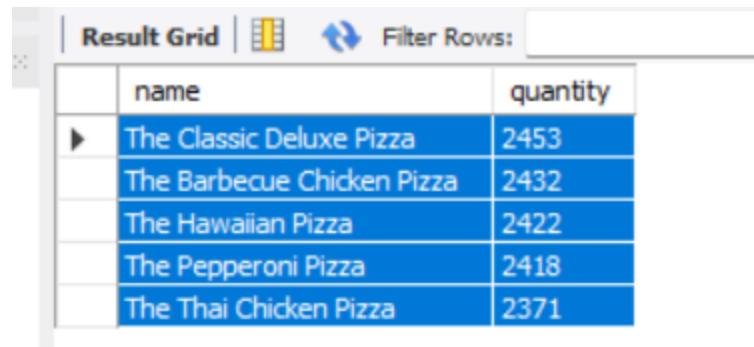


The screenshot shows a database interface with a 'Result Grid' tab. The grid contains a table with two columns: 'size' and 'order_count'. The data is sorted in descending order of 'order_count'. The first row is highlighted with a mouse cursor. Above the table, there are icons for a grid, a refresh button, and a 'Filter Rows:' label.

| | size | order_count |
|---|------|-------------|
| ▶ | L | 18526 |
| | M | 15385 |
| | S | 14137 |
| | XL | 544 |
| | XXL | 28 |

Q5. List the top 5 most ordered pizza types along with their quantities.

```
select pizza_types.name,  
sum(order_details.quantity) as quantity  
from pizza_types join pizzas  
on pizza_types.pizza_type_id = pizzas.pizza_type_id  
join order_details  
on order_details.pizza_id = pizzas.pizza_id  
group by pizza_types.name order by quantity desc limit 5;
```

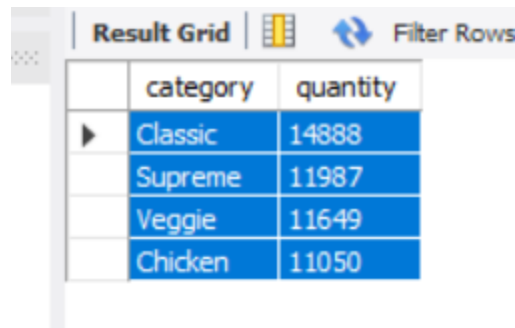


The screenshot shows a database interface with a 'Result Grid' tab. It displays the results of a SQL query, showing the top 5 most ordered pizza types by quantity. The table has two columns: 'name' and 'quantity'. The rows are sorted in descending order of quantity.

| | name | quantity |
|---|----------------------------|----------|
| ▶ | The Classic Deluxe Pizza | 2453 |
| | The Barbecue Chicken Pizza | 2432 |
| | The Hawaiian Pizza | 2422 |
| | The Pepperoni Pizza | 2418 |
| | The Thai Chicken Pizza | 2371 |

Q6. Join the necessary tables to find the total quantity of each pizza category ordered.

```
select pizza_types.category,  
sum(order_details.quantity) as quantity  
from pizza_types join pizzas  
on pizza_types.pizza_type_id = pizzas.pizza_type_id  
join order_details  
on order_details.pizza_id = pizzas.pizza_id  
group by pizza_types.category order by quantity desc;
```



The screenshot shows a 'Result Grid' window with a 'Filter Rows' button. It displays a table with two columns: 'category' and 'quantity'. The data is sorted in descending order of quantity. The categories listed are Classic, Supreme, Veggie, and Chicken.

| | category | quantity |
|---|----------|----------|
| ▶ | Classic | 14888 |
| | Supreme | 11987 |
| | Veggie | 11649 |
| | Chicken | 11050 |

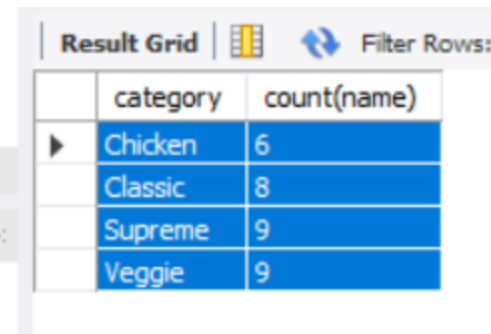
Q7. Determine the distribution of orders by hour of the day.

```
select hour(time), count(order_id) from orders  
group by hour(time);
```

| Result Grid | | | Filter Rows: |
|-------------|------------|-----------------|--------------|
| | hour(time) | count(order_id) | |
| ▶ | 11 | 1231 | |
| | 12 | 2520 | |
| | 13 | 2455 | |
| | 14 | 1472 | |
| | 15 | 1468 | |
| | 16 | 1920 | |
| | 17 | 2336 | |
| | 18 | 2399 | |
| | 19 | 2009 | |
| | 20 | 1642 | |
| | 21 | 1198 | |
| | 22 | 663 | |
| | 23 | 28 | |
| | 10 | 8 | |
| | 9 | 1 | |

Q8. Join relevant tables to find the category-wise distribution of pizzas.

```
select category, count(name) from pizza_types  
group by category;
```

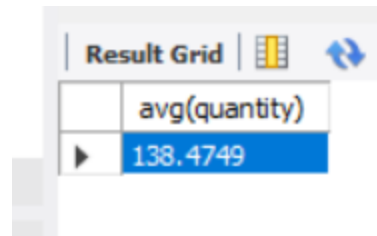


The screenshot shows a database interface with a 'Result Grid' tab. The grid contains a table with two columns: 'category' and 'count(name)'. The data is as follows:

| | category | count(name) |
|---|----------|-------------|
| ▶ | Chicken | 6 |
| | Classic | 8 |
| | Supreme | 9 |
| | Veggie | 9 |

Q9. Group the orders by date and calculate the average number of pizzas ordered per day.

```
select avg(quantity) from  
(select orders.date, sum(order_details.quantity) as quantity  
from orders join order_details  
on orders.order_id = order_details.order_id  
group by orders.date) as order_quantity ;
```



The screenshot shows a 'Result Grid' window with a single row of data. The column header is 'avg(quantity)' and the value in the row is '138.4749'. There are icons for a grid, a chart, and a refresh button at the top right of the window.

| avg(quantity) |
|---------------|
| 138.4749 |

Q10. Determine the top 3 most ordered pizza types based on revenue.

```
select pizza_types.name,  
sum(order_details.quantity * pizzas.price) as revenue  
from pizza_types join pizzas  
on pizzas.pizza_type_id = pizza_types.pizza_type_id  
join order_details  
on order_details.pizza_id = pizzas.pizza_id  
group by pizza_types.name order by revenue desc limit 3;
```

| Result Grid | | | Filter Rows: |
|-------------|------------------------------|----------|--------------|
| | name | revenue | |
| ▶ | The Thai Chicken Pizza | 43434.25 | |
| | The Barbecue Chicken Pizza | 42768 | |
| | The California Chicken Pizza | 41409.5 | |

